REPORT DOCUMENTATION PAGE

AFRL-SR-AR-TR-04-

0138

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	2/26/04	Final Technical Report, 1/1/00-11/30/03

4. TITLE AND SUBTITLE

Building Blocks for High Performance, Fault-Tolerant Distributed Systems

5. FUNDING NUMBERS

F49620-00-1-0097

6. AUTHOR(S)

Nancy Lynch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Robert Herklotz
Air Force Office of Scientific Research
4015 Wilson Blvd, Room 713
Arlington, VA 22203-1954

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

N/A

12a. DISTRIBUTION / AVAILABILITY STATEMENT

No limits of disclosure

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20040305 023

13. ABSTRACT *(Maximum 200 Words)*

This project developed theory and systems support to aid in the construction of adaptive, survivable distributed systems. The systems are designed to run in highly dynamic environments such as the internet, wireless networks, and sensor networks. Participating processes may join, leave, and fail during computation. The systems that were considered solve problems of data sharing and management, resource sharing and management, communication, and coordination.

Specifically, the project involved developing reusable "building blocks"—global service specifications and distributed algorithms—for dynamic distributed systems. The work included an extensive study of view-oriented group communication services and algorithms, which is now "transitioning" into use at Lincoln Laboratories. A major focus was on design and analysis of algorithms for implementing reliable atomic shared memory in highly dynamic networks. Other algorithmic work covered dynamic algorithms for atomic broadcast, scalable reliable multicast, and topology control.

In addition, the project produced results on mathematical semantic foundations to support modeling and analysis of highly dynamic distributed systems, and on tools to support this effort.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	31
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

Contents

1 Cover sheet **2**

2 Objectives **3**

3 Status of Effort **3**

4 Accomplishments/New Findings **3**

4.1 Algorithms 3

4.1.1 Group Communication 3

4.1.2 Reliable Broadcast 5

4.1.3 Data-Sharing Services 6

4.1.4 Peer-to-Peer Networks 8

4.1.5 Topology Control in Mobile *ad hoc* Networks 8

4.1.6 Sensor Networks 8

4.2 Modeling and Analysis Techniques 8

5 Personnel Supported **11**

6 Publications **12**

7 Interactions/Transitions **23**

7.1 Participation/presentations at meetings, conferences, seminars, etc. 23

7.2 Consultative and advisory functions to other laboratories and agencies, especially Air Force and other DoD laboratories 27

7.3 Transitions 27

8 New discoveries, inventions, patent disclosures **30**

9 Honors and awards **30**

1 Cover sheet

Final Technical Report for Agreement #F49620-00-1-0097
Building Blocks for High Performance, Fault-Tolerant Distributed Systems

Submitted to Air Force Aerospace Research-OSR, February 26, 2004

For period January 1, 2000 - November 30, 2003.

Prof. Nancy Lynch
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA

Abstract

This project developed theory and systems support to aid in the construction of adaptive, survivable distributed systems. The systems are designed to run in highly dynamic environments such as the internet, wireless networks, and sensor networks. Participating processes may join, leave, and fail during computation. The systems that were considered solve problems of data sharing and management, resource sharing and management, communication, and coordination.

Specifically, the project involved developing reusable "building blocks"—global service specifications and distributed algorithms—for dynamic distributed systems. The work included an extensive study of view-oriented group communication services and algorithms, which is now "transitioning" into use at Lincoln Laboratories. A major focus was on design and analysis of algorithms for implementing reliable atomic shared memory in highly dynamic networks. Other algorithmic work covered dynamic algorithms for atomic broadcast, scalable reliable multicast, and topology control.

In addition, the project produced results on mathematical semantic foundations to support modeling and analysis of highly dynamic distributed systems, and on tools to support this effort.

2 Objectives

This project was intended to develop theory and systems support to aid in the construction of adaptive, survivable distributed systems. The systems are designed to run in highly dynamic environments such as the internet, wireless networks, and sensor networks. Participating processes may join, leave, and fail during computation. The systems to be considered solve problems of data sharing and management, resource sharing and management, communication, and coordination.

Specifically, the project involved developing reusable "building blocks"—global service specifications and distributed algorithms—for dynamic distributed systems. The work included an extensive study of view-oriented group communication services and algorithms. A major focus was on design and analysis of algorithms for implementing reliable atomic shared memory in highly dynamic networks. Other problems to be considered included atomic broadcast, scalable reliable multicast, topology control, and clock synchronization.

In addition, the project was intended to produce results on mathematical semantic foundations to support modeling and analysis of highly dynamic distributed systems, and to produce tools to support this effort.

3 Status of Effort

The project is now completed. Many of the results obtained during this project have suggested new research directions. These are detailed in our newly-approved AFOSR proposal [27].

4 Accomplishments/New Findings

Our results fall into two categories: algorithms, and formal modeling and analysis techniques.

4.1 Algorithms

Our work on algorithms includes results on group communication, reliable broadcast, data sharing, namespace management, and topology control.

4.1.1 Group Communication

Group communication services, such as those provided by Isis [5], Transis [12], and many other systems (e.g., [31, 14, 4, 42, 19, 43]), are high-level communication services that allow processes to communicate with named groups of processes with changing membership sets. These services are based on *group membership services*, which maintain *views* of the current group membership and inform clients about changes in the views. Group communication (GC) services guarantee that communication respects views, and integrate multicast communication with the views. Within each view, they typically provide strong message ordering and reliability guarantees such as *atomic broadcast*, *causal broadcast*, or *virtual synchrony*.

Our research provided a theoretical foundation for this area. Namely, we developed precise definitions of properties to be satisfied by group communication systems, and developed methods of modeling and analyzing GC systems. We developed and analyzed new algorithms for implementing GC services and for building applications on top of GC services. We found errors in some existing implementations.

Partitionable group communication services: Fekete, Lynch, and Shvartsman [15, 5, 16] developed an I/O automaton specification for a prototype GC service patterned on services used in Isis [5] and Transis [12]. We called this service VS (for "view synchrony"). The VS service is *partitionable:* it allows disjoint views of the same group to exist simultaneously. We also developed an algorithm that uses VS to implement a *totally ordered broadcast* service based on earlier work by Keidar and Dolev [21] and by Amir et al. [1]. We proved correctness and performance properties for the algorithm, the latter conditioned on behavior of the underlying network. Archer later expanded and checked most of the correctness proofs using the PVS theorem prover [2]. We also developed computation workload distribution algorithms based on VS [22, 11]. Other researchers adopted our methods to model and analyze other GC algorithms and services [24, 7].

Working with system developers at Cornell [20], we used our methods to model and analyze Birman and Hayden's Ensemble system [43, 19]. Ensemble is organized in *layers*, where successive layers introduce successively stronger ordering and reliability properties. We developed global specifications for two key Ensemble layers—the *virtual synchrony* layer and another layer that combines virtual synchrony with a consistent total ordering property—and we modeled the Ensemble algorithm that bridges between the layers. While attempting a proof, we discovered a significant logical error in the *state exchange* sub-algorithm. This was repaired in the actual system, and we subsequently developed models and proofs for the repaired system. The same error was found to exist in Ensemble's predecessor GC system, Horus [42].

Non-partitionable services: Partitioning is undesirable for applications with strong consistency requirements, such as totally-ordered broadcast communication or coherent data management. Following an approach of Yeger Lotem et al [26], De Prisco, Fekete, Lynch, and Shvartsman defined a new GC service, DVS, which prevents partitioning, allowing only one *primary* view to exist at a time. Primary views may evolve over time, as long as each view contains members of the previous view. We also developed (and proved correctness of) an algorithm that implements DVS, and an algorithm that uses DVS to implement consistent totally ordered broadcast. Ingols implemented several primary service algorithms on a LAN and compared their performance experimentally [56].

We extended this work to allow views to contain extra structure useful for group-oriented applications, for example, a distinguished leader or sets of quorums [10, 9]. Such structure can be used, for example, to achieve consistency and availability in the face of transient changes in the set of participating processes. We call such augmented views *configurations.* We specified two configuration-oriented GC services, which allow configurations to change, provided that each configuration satisfies certain intersection properties with respect to the previous configuration, and we designed algorithms that implement our services, by extending the algorithm from [26]. Furthermore, we developed two consistent replicated data algorithms that use our services: one based on Lamport's Paxos algorithm [23], and the second based on a dynamic, multi-writer version of the

algorithm of Attiya et al. [3]. Both algorithms tolerate long-term changes in the set of participating processes (using reconfiguration) and transient changes (using quorums).

Scalable group communication: Traditional implementations of GC services have been designed for local area networks, and their communication and latency requirements do not scale well to larger networks. Khazan and Keidar developed a *scalable* GC service, intended for use in wide-area networks [31, 9, 57, 8] Their algorithm uses a separate *scalable membership service* [75], implemented on a small set of membership servers. Multicast communication, however, is implemented on all the nodes. In the new algorithm, a view change involves only one round for state exchange, and that round is conducted in parallel with the membership service's agreement on views. Moreover, new participants can join during view formation. Khazan proved correctness (safety and liveness) of the new algorithm [8], and analyzed its performance, conditioned on limitations on timing and failure behavior. In particular, he analyzed the time from when the underlying network stabilizes until the GC service announces a new view, and also analyzed message latency in stable situations. Khazan also designed, modeled, and analyzed a data-sharing application running on top of the new GCS. Tarashchanskiy implemented the new algorithm [63].

Other work: Keidar, et al. have written a comprehensive article for Computing Surveys defining and classifying the interesting guarantees provided by group communication services [8]. Other work from our research group on group communication systems appears in [41, 40, 18].

4.1.2 Reliable Broadcast

Early-delivery atomic broadcast: Bar-Joseph, Keidar, and Lynch developed a new, fast algorithm for *atomic broadcast* in a dynamic setting, where processes may join, leave, or fail [19, 67]. The problem is to guarantee that participants receive sequences of messages that are consistent with a single global message ordering; in particular, they must receive the same final message from a failed process. In the absence of failures, our algorithm guarantees constant latency, even when participants join and leave. In the presence of failures, the latency is linear in the number of failures that actually occur. The main difficulties are that the underlying network does not guarantee a single total order, and that different processes may receive different final messages from a failed process. So, processes coordinate message delivery: They divide time into slots, assign messages to slots, and deliver messages slot-by-slot. Processes determine the members of each slot, and deliver messages only from members. To decide on which processes fail in each slot, processes engage in a distributed consensus protocol. This requires a new kind of consensus service, in which participants do not know who the other participants are. We defined this new consensus problem, and developed a new *early-stopping* algorithm to solve it. Our algorithm improves upon previously-suggested algorithms using group communication.

Scalable reliable multicast: Livadas completed his PhD thesis [58], on analyzing and comparing reliable multicast protocols. He designed a new caching-enhanced version of the well-known SRM protocol, which he calls CESRM. His analysis shows that, in cases when the expedited recovery occurs, the latency is only about one fourth of that of un-enhanced SRM. By analyzing real IP

multicast traces, he has shown that expedited recoveries occur about one third of the time. This work is reported in [36, 58, 78, 37, 79, 77].

4.1.3 Data-Sharing Services

Reconfigurable atomic memory (RAMBO): Lynch and Shvartsman defined a new, reconfigurable algorithm for implementing atomic read/write shared memory in dynamic networks [43, 81], for example, mobile settings or peer-to-peer settings. The algorithm, which we call RAMBO (for "Reconfigurable Atomic Memory for Basic Objects"), tolerates short-term changes by using quorums, and tolerates long-term changes by reconfiguring. Reconfiguration occurs on-the-fly, without heavyweight view change as in group communication systems. The algorithm maintains atomicity across configuration changes.

The starting point for this algorithm is a static two-phase quorum-based implementation for read/write shared memory, as described in [28, 3]. In the first phase of a read or write operation, a value and associated tag are read from a read-quorum of replicas, and in the second phase, a value and tag are propagated to a write-quorum. A write operation uses the first phase to determine the largest tag, picks a larger tag, and uses the second phase to write the new value and tag. A read operation uses the first phase to determine the latest value and tag, and uses the second phase to propagate this information, before returning the value to its client. Operations may proceed concurrently; quorum intersection properties imply that the shared data appears to be atomic.

RAMBO adapts this strategy for use in a dynamic setting, by allowing the system to reconfigure its sets of read-quorums and write-quorums. It allows any member of the current quorum configuration to propose a new configuration; conflicts are resolved using a distributed consensus algorithm such as Paxos [23]. Although consensus is a heavyweight mechanism, it is used here only for reconfiguration, which is an infrequent operation. Also, reconfigurations do not significantly delay read and write operations, unlike what happens in group-communication-based approaches.

A process conducting a read or write operation runs an algorithm similar to the static two-phase algorithm described above, using the current configuration. When a new configuration is determined, the read or write operation continues, using the new configuration in addition to the old one; this may require additional work to access processes needed for new quorums. An old configuration may be abandoned, after execution of a two-phase "garbage-collection" procedure. Garbage-collection is performed in the background, concurrently with read and write operations. We have proved that, under "normal" timing assumptions, garbage-collection of old configurations keeps up with introduction of new configurations, and read and write operations take time at most $8d$, where d is a bound on the message delay. Musial and Shvartsman have built a preliminary implementation of RAMBO in a LAN [47], and are beginning to carry out experiments.

Rambo II: The RAMBO algorithm garbage-collects old quorum configurations sequentially, which works well under normal timing assumptions. However, if normal timing assumptions are violated, configurations can pile up, and garbage-collection may take a long time to catch up. Recently, Gilbert, Lynch, and Shvartsman have improved RAMBO with a new garbage-collection procedure, which handles the removal of any number of old configurations in parallel [26, 72].

Separating data and metadata: Many quorum-based data-sharing algorithms (e.g., [3, 28, 43, 81]) couple data closely to metadata such as tags, for example, sending data and tags together in messages. However, in practice, metadata information may be much smaller than the actual data. Fan [23, 24, 53] has developed a variation on the standard quorum-based read/write algorithms that separates the handling of the metadata from that of the actual data. Most of the complicated processing involves the metadata only. Handling of the actual data is quite straightforward and flexible: all that is necessary is to maintain enough copies of the right versions, and to make them easy to find. The algorithm is efficient. For example, in each logical read operation, only one physical read of the data and no physical writes are performed. This is a substantial improvement over most other quorum-based atomic data algorithms, which read and write data to two quorums during a logical read. Each read or write operation uses three phases. While this is more than the two phases required in some other algorithms [3, 28, 43, 81], some of the phases send only short messages, and so should take negligible time in practice. Fan's algorithm is designed for a static environment. But because it allows data to be stored in any (large enough) set of processes, and not only in quorums, it supports reconfiguration of replicas in response to failures or changing network conditions. Furthermore, the reconfiguration is cheap enough to be applied on a per-operation basis. Fan has also proved two lower bounds that show that some of the costs incurred by his algorithm are necessary. In particular, he has proved lower bounds on the number of replicas that are written by read operations, and on the number of versions of an object that must be maintained.

GeoQuorums: Dolev, Gilbert, Lynch, Shvartsman, and Welch have developed a new approach to maintaining atomic data in mobile *ad hoc* networks, or in networks that consist of a combination of mobile nodes and fixed nodes [21, 69, 4]. This approach involves building an abstract layer consisting of virtual *focal point processes*, each associated with a fixed geographical region. We believe that introducing such a layer provides a viable, practical method of programming mobile ad hoc networks.

In the GeoQuorums algorithm, the focal point processes engage in a (mostly static) quorum-based algorithm to implement atomic read/write objects. This algorithm includes certain optimizations over the two-phase protocols used in [28, 3, 43, 81]: the first phase of a write is omitted in favor of a tag-selection protocol based on synchronized local clocks, and the second phase of a read is omitted when the operation learns that the relevant information has already been propagated by another operation. Also, quorums are chosen for good performance in the geographical setting. Focal point processes are implemented using either fixed or mobile nodes; in the case of mobile nodes, replicated state machine techniques, based on local broadcast, are used.

Brewer's conjecture: Gilbert and Lynch [6] have proved several versions of a "folklore" result stated a few years ago by Eric Brewer: that it is impossible, in a fault-prone communication network, to guarantee a combination of atomicity (consistency) and availability. The precise statement of this result depends on assumptions about timing and failures; the paper [6] obtains results for several sets of assumptions.

4.1.4 Peer-to-Peer Networks

Atomic shared memory: At the first International Workshop on Peer-to-Peer systems [33], Lynch, Malkhi, and Ratajczak [44] proposed a technique for implementing atomic read/write memory in ring-like peer-to-peer networks like the classical CHORD network [39]. Our method integrates object-management facilities directly into the basic namespace-management protocols, by systematically passing control of objects to new processes when the ring changes.

MultiChord: Lynch and Stoica [82] designed and evaluated a peer-to-peer namespace management service called MULTICHORD [82]. This algorithm, based on CHORD [39], is tuned for performance in a *steady state* situation, where clients join and fail at a bounded rate. In particular, (1) each node maintains table entries for several neighbors of each power-of-two successor in the namespace, as well as for its own neighbors, and (2) each node delays joining until its table has been populated with all the required entries.

4.1.5 Topology Control in Mobile *ad hoc* Networks

Preserving mulltiple connectivity: Finally, several PhD students working with our group have been studying the problem of controlling network topology in mobile *ad hoc* networks, by adjusting the nodes' power settings [16, 27]. In [16], they presented a distributed algorithm for adjusting power while preserving k-connectivity of the network communication graph, using "cone-based" techniques based on work of Li et al. [25]. They also proved optimality results for their choice of cone sizes, and extended the results to three dimensions. Preserving k-connectivity is interesting because a k-connected graph includes enough redundancy to support message routing in the face of $k-1$ node failures. In [27], they developed a measure of total power consumption and developed several algorithms whose total power consumption approximates the global optimum.

4.1.6 Sensor Networks

We began work on algorithms for networks of sensors, focusing on problems of time synchronization, message dissemination, tracking, and routing. We are currently writing papers on new algorithms for tracking and time synchronization.

4.2 Modeling and Analysis Techniques

Our group has a long history of developing fundamental modeling frameworks for distributed systems, based on interacting state machines (various forms of I/O automata).

Hybrid I/O Automata: Recent accomplishments in this direction include a comprehensive development and presentation of the *Hybrid I/O Automata (HIOA)* modeling framework [10], by Lynch, Segala, and Vaandrager. This framework supports description and analysis of *hybrid systems*, which may consist of real-world components such as land-based vehicles or airplanes, as well as computer components. The framework allows description of continuous state evolution

as well as discrete transitions. Hybrid systems can be composed to build larger systems, with continuous as well as discrete interaction between components, and can be modeled at multiple levels of abstraction. We have used the HIOA framework for many case studies in the area of transportation control, including studies of the TCAS collision-avoidance system [34], of a Quanser model helicopter system [46], and of a circumvention and recovery system for an inertial guidance system [55]. We have also used HIOA to describe mobile computing systems, in [4, 70].

Timed I/O Automata: Kaynar, Lynch, Segala, and Vaandrager are currently completing a comprehensive monograph on the *Timed I/O Automata (TIOA)* modeling framework, intended to support the description and analysis of timed systems. An important feature of this model is its support for decomposing timed system descriptions. In particular, the framework includes a notion of external behavior for a timed I/O automaton, which captures its discrete interactions with its environment. The framework also defines what it means for one TIOA to implement another, based on an inclusion relationship between their external behaviors, and defines notions of simulations, which provide sufficient conditions for demonstrating implementation relationships. The framework includes a composition operation for TIOAs, which respects external behavior, and a notion of receptiveness, which implies that a TIOA does not "block" the passage of time. The TIOA framework supports the statement and verification of safety and liveness properties for timed systems. It defines what it means for a property to be a safety or a liveness property, includes basic results about safety/liveness classification, and receptiveness for liveness properties.

The TIOA framework is formalized as a special case of the HIOA framework, in which components interact by discrete events only [73]. This restriction leads to simplifications in some of the HIOA definitions and results, especially those involving composition. Although this monograph contains some new results (e.g., about liveness properties) its main purpose is to serve as a "user manual" to explain practical methods of modeling and analyzing timing-based distributed systems. A summary version appeared in RTSS'03 [30].

Probabilistic I/O Automata: Lynch, Segala, and Vaandrager have begun working on compositional models of systems that include probabilistic behavior, based on earlier definitions of *Probabilistic I/O Automata (PIOA)* by Segala [35, 36, 37] (also see Stoelinga [38]). In a preliminary paper [45], we have characterized directly, in terms of a simulation relation, a notion of implementation for PIOAs that was previously defined only implicitly [35, 36, 37]. Perhaps surprisingly, this turns out to be a very fine relation, exposing much of the internal branching structure of the automata. We are currently working on developing coarser implementation relations, based on restricting the scheduling of system components.

Dynamic I/O Automata: Finally, Attie and Lynch [65] have developed a *Dynamic I/O Automata* modeling framework, which allows component capabilities (signatures) to change and includes explicit actions representing process creation and destruction [66]. This work is somewhat related to Milner's Pi-Calculus [29, 30]; however, our specific choices of primitive notions to include in the model are quite different, and our focus is on mathematical semantics rather than on formal notation and process calculi. Early versions of this work appeared in [15, 14].

Analysis methods: As a byproduct of our work on scalable group communication, described in Section 4.1.1, we found new methods of *incremental modeling and proof* for distributed system components modeled as I/O automata [8]. These techniques are inspired by inheritance notions from object-oriented programming. These techniques support incremental construction of specifications, algorithm descriptions, and abstraction proofs showing that algorithms meet their specifications. We used these methods to develop models and safety proofs for our scalable GC services.

We analyzed the performance of several complex algorithms using conditional methods [5, 57, 43, 81, 26, 72, 82]. In particular, we proved latency bounds in situations in which the underlying system exhibits completely benign behavior from some point onward [5, 57, 82]. We also proved bounds for situations in which the system exhibits good *steady-state* behavior throughout the execution [43, 81, 82]—not completely benign behavior, but rather, behavior in which the rate of change (joins and failures) is bounded. We also analyzed some protocols in situations in which timing and failure behavior is arbitrary up to some point, then follows a steady-state pattern from that point onward [26, 72]. Steady-state analysis is still difficult to do; more remains to be done to make these methods tractable.

Tools: Garland, Kaynar, and Lynch, working with many students, have been designing, implementing, and testing the experimental *IOA specification language and toolset.* This language and toolset are intended to support distributed programming based on high-level, composable mathematical models. Tools include:

1. A simulator, which is capable of simulating algorithms using multiple levels of abstraction. Early versions of the simulator were designed and built by Chefter [6], Ramirez [61], and Dean [52]. The latest version of the simulator, by Solovey [62], is also capable of simulating the composition of several automata. The latest reference manual for the simulator appears in [74].
2. A connection to the Daikon invariant discovery tool [13].
3. A connection to the Larch theorem-prover [17, 51, 20].
4. A connection to the Isabelle/HOL theorem-prover [48, 60]. Ne Win used this connection for experiments in reducing the amount of human interaction required to discover and prove interesting properties of distributed algorithms [11]. These experiments used invariants suggested by Daikon.
5. A "Composer" tool, an addition to the IOA front end, which expands the definition of a composite automaton into an equivalent primitive automaton [100].
6. A code generator for distributed code (to run in our LAN) is still in progress [98, 99].

The latest release of the IOA language and toolset is available at URL `http://theory.lcs.mit.edu/tds/ioa` It includes a comprehensive user guide and reference manual [97]

These tools do not (yet) support additions to the model such as timing, continuous behavior, or probabilistic behavior. We are currently extending the IOA language with notations for timing behavior based on the theoretical framework in [73], and plan to extend the tools to accommodate this extension.

5 Personnel Supported

Prof. Nancy Lynch

Postdoctoral associates:

Gregory Chockler
Idit Keidar
Dilsun Kirli Kaynar

PhD students:

Rui Fan (received MS in 2/03)
Seth Gilbert (received MS in 8/03)
Roger Khazan (graduated in 5/02)
Carolos Livadas (graduated in 7/03)
Victor Luchangco (graduated in 9/01)
Sayan Mitra
Tina Nolte
Joshua Tauber

MEng students:

Omar Bakr (graduated in 1/03)
Andre Bogdanov (graduated in 9/01)
Laura Dean (graduated in 9/01)
Vida Uyen Ha (5/03)
Kyle Ingols (5/00)
Toh Ne Win (graduated in 6/03) J. Antonio Ramırez-Robredo (graduated in 9/00)
Christine Robson (current)
Ed Solovey (graduated in 8/03)
Igor Tarashchanskiy (graduated in 9/00) Michael Tsai (graduated in 6/02)

UROPs (IOA): Gustavo Santos, Chris Luhrs, Shien Jin Ong, Atish Nigam, Stan Funiak

Group affiliates and visitors: Paul Attie (Northeastern U.)
Roberto De Prisco (U. Salerno)
Shlomi Dolev (Ben Gurion U.)
Stephen Garland
Yoshinobu Kawabe (NTT)
Roberto Segala (U. Verona)
Alex Shvartsman (U. Conn.)
Frits Vaandrager (U. Nijmegen)
Jennifer Welch (Texas A&M)

6 Publications

Journal Publications

[1] P. C. Attie. Wait-free Byzantine Consensus. *Information Processing Letters*, vol. 83, no. 4, pp. 221-227, August 2002.

[2] Bogdan S. Chlebus, Roberto De Prisco, and Alex A. Shvartsman. Performing tasks on synchronous restartable message-passing processors. *Distributed Computing*, 14:49-64, 2001.

[3] Roberto DePrisco, Butler Lampson, and Nancy Lynch. Fundamental Study: Revisiting the PAXOS algorithm. *Theoretical Computer Science*, 243:35-91, 2000.

[4] Shlomi Dolev, Seth Gilbert, Nancy Lynch, Alex Shvartsman, and Jennifer Welch. GeoQuorums: Implementing atomic memory in ad hoc networks. Selected for special edition of *Distributed Computing*, (edited by Faith Fich), related to the *DISC03 conference*, 2004. Also, to appear as Technical Report MIT-LCS-TR-900a, CSAIL, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[5] Alan Fekete, Nancy Lynch, and Alex Shvartsman. Specifying and Using a Partitionable Group Communication Service. *ACM Transactions on Computer Systems*, 19(2):171-216, May 2001.

[6] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *Sigact News*, 33(2), June 2002.

[7] Idit Keidar and Sergio Rajsbaum. A simple proof of the uniform consensus synchronous lower bound. *Information Processing Letters (IPL)*. 85(1):47-52, January 2003.

[8] Idit Keidar, Roger Khazan, Nancy Lynch, and Alex Shvartsman. An Inheritance-Based Technique for Building Simulation Proofs Incrementally. *ACM Transactions on Software Engineering and Methodology (TOSEM)*. 11(1):1-29, January 2002.

[9] Idit Keidar and Roger Khazan. A virtually synchronous group multicast algorithm for WANs: Formal approach. *SIAM Journal on Computing*, 23(1):78-130, 2002.

[10] Nancy Lynch, Roberto Segala, and Frits Vaandraager. Hybrid I/O Automata. *Information and Computation*, 185(1):105-157, August 2003. Also, Technical Report MIT-LCS-TR-827b, MIT Laboratory for Computer Science Technical Report, Cambridge, MA 02139, February 2002.
`theory.lcs.mit.edu/tds/papers/Lynch/HIOA-final.ps`.

[11] Toh Ne Win, Michael Ernst, Stephen Garland, Dilsun Kirli, and Nancy Lynch. Using simulated execution in verifying distributed algorithms. To appear in *International Journal on Software Tools for Technology Transfer (STTT)*.

Reviewed Conference Proceedings

[12] Tadashi Araragi, Paul Attie, Idit Keidar, Kiyoshi Kogure, Victor Luchangco, Nancy Lynch, and Ken Mano. On Formal Modeling of Agent Computations. In *NASA Workshop on Formal Approaches to Agent-Based System*, April, 2000.

[13] Paul C. Attie. On the Implementation Complexity of Specifications of Concurrent Programs. *Distributed Computing (DISC 2003: 17th International Symposium on Distributed Computing, Sorrento, Italy, October 2003)*, volume 2848 of Lecture Notes in Computer Science, pages 151-165, Springer-Verlag, 2003.

[14] Paul Attie and Nancy Lynch. Dynamic Input/Output automata: A formal model for dynamic systems. In K. G. Larsen and M. Nielsen, editors, *CONCUR 2001 - Concurrency Theory: 12th International Conference, Aaalborg, Denmark, August 20-25, 2001, Proceedings.*, volume 2154 of *Lecture Notes in Computer Science*, pages 137–151. Springer-Verlag, 2001. Also, Technical Report, College of Computing, Northeastern University, January 2001.

[15] Paul C. Attie and Nancy A. Lynch. Brief Announcement: Dynamic Input/Output Automata: A Formal Model for Dynamic Systems. *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing*, Newport, RI, pages 314-316, August 2001.

[16] M. Bahramgiri, M. T. Hajiaghayi, and V. S. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks. In *Proceedings of the 11th IEEE International Conference on Computer Communications and Networks (IC3N)*, pages 392–398, Miami, Floria, October 2002. Also, MIT Technical Report MIT-LCS-TR-862, Cambridge, MA 02139, 2002.

[17] Omar Bakr and Idit Keidar. Evaluating the running time of a communication round over the internet. In *21st ACM Symposium on Principles of Distributed Computing (PODC '02)*, pages 243–252, Monterey, CA, USA, July 2002.

[18] Ziv Bar-Joseph, Idit Keidar, Tal Anker, and Nancy Lynch. QoS Preserving Totally Ordered Multicast. In the *5th International Conference On Principles Of DIstributed Systems (OPODIS)*, pages 143-162, Paris, France, December, 2000.

[19] Ziv Bar-Joseph, Idit Keidar, and Nancy Lynch. Early-delivery dynamic atomic broadcast. In D. Malkhi, editor, *Distributed Computing (Proceedings of the 16th International Symposium on DIStributed Computing (DISC) October 2002, Toulouse, France)*, volume 2508 of Lecture Notes in Computer Science, pages 1-16, 2002. Springer-Verlag.

[20] Andrej Bogdanov, Stephen Garland, and Nancy Lynch. Mechanical translation of I/O automaton specifications into first-order logic. In In Doron Peled, Moshe Y. Vardi, editors, *Formal Techniques for Networked and Distributed Systems - FORTE 2002 (Proceedings of the 22nd IFIP WG 6.1 International Conference, Houston, Texas, USA, November 11-14, 2002)*, volume 2529 of Lecture Notes in Computer Science, pages 364-368, Springer 2002.

[21] Shlomi Dolev, Seth Gilbert, Nancy Lynch, Alex Shvartsman, and Jennifer Welch. GeoQuorums: Implementing atomic memory in ad hoc networks. *Distributed Computing (DISC 2003: 17th International Symposium on Distributed Computing, Sorrento, Italy, October, 2003)*, volume 2848 of Lecture Notes in Computer Science, pages 306-320, Springer-Verlag, 2003.

[22] B. Englert, L. Rudolph, and A.A. Shvartsman. Developing and refining an adaptive token-passing strategy. In *Proceedings of the IEEE International Conference on Distributed Computer Systems (ICDCS'2001)*, 10 pages, 2001.

[23] Rui Fan and Nancy Lynch. Efficient replication of large data objects. In *Proceedings of the Twenty-Second Annual ACM Symposium on Principles of Distributed Computing*, Boston, Massachusetts, July 2003.

[24] Rui Fan and Nancy Lynch. Efficient replication of large data objects. *Distributed Computing (DISC 2003: 17th International Symposium on Distributed Computing, Sorrento, Italy, October, 2003)*, volume 2848 of *Lecture Notes in Computer Science*, pages 75-91, Springer-Verlag, 2003.

[25] Alan Fekete and Idit Keidar. A Framework for Highly Available Services Based on Group Communication. In the *IEEE International Workshop on Applied Reliable Group Communication (WARGC)*, held in conjunction with *ICDCS 2001*, pages 57-62, Phoenix, Arizona, April 2001.

[26] Seth Gilbert, Nancy Lynch, and Alex Shvartsman. RAMBO II: Rapidly reconfigurable atomic memory for dynamic networks. In *International Conference on Dependable Systems and Networks*, pages 259–268, San Francisco, CA, June 2003.

[27] Mohammad Taghi Hajiaghayi, Nicole Immorlica, and Vahab S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. *MOBICOM 2003: Proceedings of the Ninth Annual ACM International Conference on Mobile Computing and Networking*, pages 300-312, San Diego, CA, September 2003.

[28] Kyle Ingols and Idit Keidar. Availability Study of Dynamic Voting Algorithms. In the *21st International Conference on Distributed Computing Systems (ICDCS)*, pages 247-254, Phoenix, Arizona, April 2001. Previous version: MIT Technical Memorandum MIT-LCS-TM-611, November 2000.

[29] Dilsun Kirli Kaynar, Anna Chefter, Laura Dean, Stephen J. Garland, Nancy A. Lynch, Toh Ne Win, and Antonio Ramírez-Robredo. Simulating nondeterministic systems at multiple levels of abstraction. In *Proceedings of Tools Day affliated to CONCUR 2002*, Brno, Czech Republic, August 2002. Also avaliable as a Technical Report of the Faculty of Informatics, Masaryk University, Czech Republic. FIMU-RS-2002-05.

[30] Dilsun K. Kaynar, Nancy Lynch, Roberto Segala, and Frits Vaandrager. A framework for modeling timed systems with restricted hybrid automata. *RTSS 2003: The 24th IEEE International Real-Time Systems Symposium*, Cancun, Mexico, December 2003.

[31] Idit Keidar and Roger Khazan. A Client-Server Approach to Virtually Synchronous Group Multicast: Specifications and Algorithms. *IEEE 20th International Conference on Distributed Computing Systems (ICDCS)*, pages 344-355, Taipei, Taiwan, April 2000.

[32] I. Keidar and K. Marzullo. The need for realistic failure models in protocol design. In *4th International Survivability Workshop (ISW) 2001/2002*, Vancouver, Canada, March 2002.

[33] Idit Keidar. Challenges in evaluating distributed algorithms. In *Future Directions in Distributed Computing (FuDiCo 2002, Bertinoro, Italy, June 2002)*, volume 2584 of *Lecture Notes in Computer Science*, pages 40-44, 2003. Springer.

[34] Idit Keidar, Roger Khazan, Nancy Lynch, and Alex Shvartsman. An Inheritance-Based Technique for Building Simulation Proofs Incrementally. *22nd International Conference on Software Engineering (ICSE)*, pages 478-487, Limerick, Ireland, June 2000.

[35] Roger Khazan and Nancy Lynch. An Algorithm for an Intermittently Atomic Data Service Based on Group Communication. *International Workshop on Large-Scale Group Communication*, pages 25-30, Florence, Italy, October 2003.

[36] Carolos Livadas, Idit Keidar, and Nancy A. Lynch. Designing a Caching-Based Reliable Multicast Protocol. *Proceedings of the International Conference on Dependable Systems and Networks (DSN'01)*, Fast Abstracts Supplement, B44-B45, Gothenburg, Sweden, July 2001.

[37] Carolos Livadas and Nancy A. Lynch. A Formal Venture into Reliable Multicast Territory. In Doron Peled, Moshe Y. Vardi, editors, *Formal Techniques for Networked and Distributed Systems - FORTE 2002 (Proceedings of the 22nd IFIP WG 6.1 International Conference, Houston, Texas, USA, November 11-14, 2002)*, volume 2529 of *Lecture Notes in Computer Science*, pages 146-161, Springer 2002.

[38] Victor Luchangco. Modeling Weakly Consistent Memories with Locks. *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 332-333, Crete, Greece, July 2001.

[39] Nancy Lynch and Alex Shvartsman. Communication and Data Sharing for Dynamic Distributed Systems. In A. Schiper, A.A. Shvartsman, H. Weatherspoon, B.Y. Zhao, editors, *Future Directions in Distributed Computing: Research and Position Papers (FuDiCo 2002: Proceedings of the International Workshop on Future Directions in Distributed Computing, Bertinoro, Italy, June 2002)*, volume 2584 of Lecture Notes in Computer Science, pages 62-67, Springer-Verlag, 2003

[40] Nancy Lynch. Some Perspectives on PODC. *Distributed Computing*, 16(1):71-74, 2003.

[41] Nancy Lynch. Working with Mike on Distributed Computing Theory, 1978-1992. *Proceedings of the Twenty-Second Annual ACM Symposium on Principles of Distributed Computing (PODC'03)*, page 11, Boston, MA, July 2003.

[42] Nancy Lynch, Roberto Segala, and Frits Vaandrager. Hybrid I/O Automata Revisited. In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vincentelli, editors Hybrid Systems: *Computation and Control. Fourth International Workshop HSCC'01*, Rome, Italy, March 2001, volume 2034 of *Lecture Notes in Computer Science*, pages 403-417, 2001. Springer-Verlag.

[43] Nancy Lynch and Alex Shvartsman. RAMBO: A reconfigurable atomic memory service for dynamic networks. In D. Malkhi, editor, *Distributed Computing (Proceedings of the 16th International Symposium on DIStributed Computing (DISC), Toulouse, France, October 2002)*, volume 2508 of *Lecture Notes in Computer Science*, pages 173–190. Springer-Verlag, 2002.

[44] Nancy Lynch, Dahlia Malkhi, and David Ratajczak. Atomic data access in content addressable networks. In P. Druschel, F. Kaashoek, and A. Rowstron, editors, *Peer-to-Peer Systems (First International Workshop on Peer-to-Peer Computing, Cambridge, MA, March 2002)*, volume 2429 of Lecture Notes in Computer Science, pages 295-305, Springer, 2002.

[45] Nancy Lynch, Roberto Segala, and Frits Vaandrager. Compositionality for probabilistic automata. In Roberto Amadio and Denis Lugiez, editors, *CONCUR 2003 - Concurrency Theory (14th International Conference on Concurrency Theory, Marseille, France, September, 2003)*, volume 2761 of *Lecture Notes in Computer Science*, pages 208-221, Springer-Verlag, 2003. Also, long version to appear as Technical Report MIT-LCS-TR-907, MIT Laboratory for Computer Science, Cambridge, MA 02139.

[46] Sayan Mitra, Yong Wang, Nancy Lynch, and Eric Feron. Safety Verification of Model Helicopter Controller using Hybrid Input/Output Automata. O. Maler, A. Pnueli, editors, *Hybrid Systems: Computation and Control (6th International Workshop, HSCC'03, Prague, the Czech Republic April 3-5, 2003)*, volume 2623 of *Lecture Notes in Computer Science*, pages 343-358, Springer-Verlag, 2003.

[47] P.M. Musial and A.A. Shvartsman Implementing a Reconfigurable Atomic Memory Service for Dynamic Networks. To appear in the *9th IEEE Workshop on Fault-Tolerant Parallel, Distributed and Network-Centric Systems*, 2004.

[48] Toh Ne Win, Michael D. Ernst, Stephen J. Garland, Dilsun K. Kaynar, and Nancy Lynch. Using simulated execution in verifying distributed algorithms. L.D. Zuck, P.C. Attie, A. Cortesi, S. Mukhopadhyay, editors, *Verification, Model Checking, and Abstract Interpretation (Proceedings of 4th International Conference, VMCAI 2003, New York, NY, USA, January 9-11, 2003)*, volume 2575 of *Lecture Notes in Computer Science*, pages 283-297, Springer-Verlag, 2003

[49] Jeremy Sussman, Idit Keidar, and Keith Marzullo. Optimistic Virtual Synchrony. *19th IEEE Symposium on Reliable Distributed Systems (SRDS)*, pages 42-51, October 2000.

Theses

[50] Omar Bakr. Performance Evaluation of Distributed Algorithms over the Internet. Master of Engineering in Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, February 2003.

[51] Andrej Bogdanov. Formal verification of simulations between I/O automata. Master of Engineering thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, September 2001.

[52] Laura G. Dean. Improved Simulation of Input/Output Automata. Master of Engineering Thesis, Massachusetts Institute of Technology, Cambridge, MA, September 2001.

[53] R. Fan. Efficient replication of large data objects. Masters Thesis. Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139, February 2003.

[54] Seth Gilbert. RAMBO II: Rapidly Reconfigurable Atomic Memory for Dynamic Networks. Masters Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, August 2003.

[55] Vida Uyen Ha. Verification of an Attitude Control System. Bachelor of Science and Master of Engineering, Department of Electrical Engineering and Computer Science, Cambridge, MA, May 2003.

[56] Kyle W. Ingols. Availability Study of Dynamic Voting Algorithms. M.Eng. thesis, MIT Department of Electrical Engineering and Computer Science, Cambridge, MA, May 5, 2000.

[57] Roger Khazan. *A One-Round Algorithm for Virtually Synchronous Group Communication in Wide Area Networks.* PhD thesis, MIT Department of Electrical Engineering and Computer Science, Cambridge, MA, USA, May 2002.

[58] Carolos Livadas. *Formally Modeling, Analyzing, and Designing Network Protocols—A Case Study on Retransmission-Based Reliable Multicast –Protocols.* PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, July 2003.

[59] Victor Luchangco. *Memory Consistency Models for High Performance Distributed Computing.* Phd Thesis, Massachusetts Institute of Technology, Cambridge, MA, September 2001.

[60] Toh Ne Win. Theorem-proving distributed algorithms with dynamic analysis, May 2003. Master of Engineering Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

[61] J. Antonio Ramırez-Robredo. Paired simulation of I/O automata. Master of Engineering and Bachelor of Science in Computer Science and Engineering Thesis, Massachusetts Institute of Technology, Cambridge, MA, September 2000.

[62] Edward Solovey. Simulation of composite I/O automata. Masters of Engineering Thesis, MIT Department of Electrical Engineering and Computer Science, Cambridge, MA, August 2003.

[63] Igor Tarashchanskiy. Virtual Synchrony Semantics: Client-Server Implementation. Masters thesis. MIT Department of Electrical Engineering and Computer Science, Cambridge, MA, September, 2000.

[64] Michael J. Tsai. Code Generation for the IOA Language. Master of Engineering Thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2002

Technical Reports

[65] Paul C. Attie and Nancy A. Lynch. Dynamic Input/Output automata: A formal model for dynamic systems. Technical Report MIT-LCS-TR-902, MIT Laboratory for Computer Science, Cambridge, MA 02139, 2003. Also, Technical Report, College of Computer Science, Northeastern University, July 2003.

[66] Paul Attie, Nancy Lynch, and Sergio Rajsbaum. Boosting Fault-Tolerance in Asynchronous Message Passing Systems is Impossible. Technical Report MIT-LCS-TR-877, MIT Laboratory for Computer Science, Cambridge, MA, December 2002.

[67] Ziv Bar-Joseph, Idit Keidar, and Nancy Lynch. Early-delivery dynamic atomic broadcast. Technical Report MIT-LCS-TR-840, MIT Laboratory for Computer Science, Cambridge, MA, April 2002.

[68] Roberto De Prisco, Alan Fekete, Nancy Lynch, and Alex Shvartsman. A Dynamic Primary Configuration Group Communication Service. Technical Memo MIT-LCS-TR-873, MIT Laboratory for Computer Science, Cambridge, MA, November 2002.

[69] Shlomi Dolev, Seth Gilbert, Nancy Lynch, Alex Shvartsman, and Jennifer Welch. GeoQuorums: Implementing atomic memory in ad hoc networks. Technical Report MIT-LCS-TR-900, MIT Laboratory for Computer Science, Cambridge, MA, 2003.

[70] Shlomi Dolev, Seth Gilbert, Nancy Lynch, Alex Shvartsman, and Jennifer Welch. GeoQuorums: Implementing atomic memory in ad hoc networks. Also, to appear as Technical Report MIT-LCS-TR-900a, CSAIL, Massachusetts Institute of Technolgy, Cambridge, MA, 2004.

[71] Rui Fan. Efficient replication of large data-objects. Technical Report MIT-LCS-TR-886, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, February 2003.

[72] Seth Gilbert, Nancy Lynch, and Alex Shvartsman. RAMBO II: Implementing atomic memory in dynamic networks, using an aggressive reconfiguration strategy. Technical Report LCS-TR-890, Massachusetts Institute Technology, 2003.

[73] Dilsun K. Kaynar and Nancy Lynch, Roberto Segala, and Frits Vaandrager. Theory of timed I/O automata. Technical Report MIT-LCS-TR-917, MIT Laboratory for Computer Science, Cambridge, MA, 2003

[74] Dilsun Kirli Kaynar, Anna Chefter, Laura Dean, Stephen Garland, Nancy Lynch, Toh Ne Win, and Antonio Ramirez-Robredo. The IOA Simulator. Technical Report MIT-LCS-TR-843, MIT Laboratory for Computer Science, Cambridge, MA, July 2002.

[75] Idit Keidar, Jeremy Sussman, Keith Marzullo, and Danny Dolev. Moshe: A Group Membership Service for WANs. Technical Memorandum MIT-LCS-TM-593a Massachusetts Institute of Technology, Laboratory for Computer Science. also Technical Report CS99-623a University of alifornia, San Diego, Department of Computer Science and Engineering, June 1999, revised September 2000. Submitted for journal publication. Preliminary version appeared in the *20th International Conference on Distributed Computing Systems (ICDCS)*, pages 356-365, April 2000.

[76] Idit Keidar and Sergio Rajsbaum. On the Cost of Fault-Tolerant Consensus When There Are No Faults - A Tutorial. Technical Report MIT-LCS-TR-821 MIT Laboratory for Computer Science, May 24, 2001. Preliminary version in *SIGACT News* 32(2), Distributed Computing column, pages 45-63, June 2001 (published May 15th, 2001). Partially submitted for journal publication.

[77] Carolos Livadas and Nancy Lynch. A reliable broadcast scheme for sensor networks. Technical Report MIT-LCS-TR-915, MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, August 2003.

[78] Carolos Livadas and Idit Keidar. The Case for Exploiting Packet Loss Locality in Multicast Loss Recovery. Technical Report MIT/LCS/TR-867, MIT Laboratory for Computer Science, Cambridge, MA, October 2002.

[79] Carolos Livadas and Nancy A. Lynch. A Formal Venture into Reliable Multicast Territory. Technical Report MIT-LCS-TR-868, MIT Laboratory for Computer Science, Cambridge, MA, November 2002.

[80] Nancy Lynch, Roberto Segala, and Frits Vaandraager. Hybrid I/O Automata. Technical Report MIT-LCS-TR-827d, MIT Laboratory for Computer Science, Cambridge, MA 02139, January 13, 2003. (Earlier versions in 2001).

[81] Nancy Lynch and Alex Shvartsman. RAMBO: A Reconfigurable Atomic Memory Service for Dynamic Networks. Technical Report MIT-LCS-TR-856, MIT Laboratory for Computer Science, Cambridge, MA, August 2002.

[82] Nancy Lynch and Ion Stoica. MultiChord: A resilient namespace management algorithm. To appear as Technical Memo MIT-LCS-TR-936, CSAIL, Massachusetts Institute of Technology, Cambridge, MA 2004.

[83] Sayan Mitra, Yong Wang, Nancy Lynch, and Eric Feron. Application of Hybrid I/O Automata in Safety Verification of Pitch Controller for Model Helicopter System. Technical Report MIT-LCS-TR-880, MIT Laboratory for Computer Science, Cambridge, MA 02139, January 2003.

Manuscripts

[84] Roberto De Prisco, Alex Shvartsman, Nicole Immorlica, and Toh Ne Win. A Formal Treatment of Lamport's Paxos Algorithm. Manuscript, 2002.

[85] Stephen J. Garland and Nancy A. Lynch and Mandana Vaziri. IOA: A Language for Specifying, Programming and Validating Distributed Systems. User and Reference Manual. Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, January 2001.

Submitted for Publication

[86] Paul Attie, Rachid Guerraoui, Petr Kouznetsov, Nancy Lynch, and Sergio Rajsbaum. Boosting Distributed Service Resilience is Impossible. Submitted for publication.

[87] Michael A. Bender, Jeremy T. Fineman, Seth Gilbert, Charles E. Leiserson. On-the-Fly Maintenance of Series-Parallel Relationships in Fork-Join Multithreaded Programs. Submitted for publication.

[88] Jacob Beal and Seth Gilbert. RamboNodes for the Metropolitan Ad Hoc Network. Submitted for publication. Also, AI Memo: AIM-2003-027.

[89] Murat Demirbas, Anish Arora, Tina Nolte, and Nancy Lynch. STALK: A Self-Stabilizing Hierarchical Tracking Service for Sensor Networks. Submitted for publication.

[90] Shlomi Dolev, Seth Gilbert, Nancy A. Lynch, Elad Schiller, Alex A. Shvartsman, and Jennifer Welch. Virtual Mobile Nodes for Mobile Ad Hoc Networks. Submitted for publication. Also, to appear as Technical Report MIT-LCS-TR-937, MIT CSAIL, Cambridge, MA, 2004.

[91] Rui Fan and Nancy Lynch. Gradient Clock Synchronization. Submitted for publication.

[92] Sayan Mitra and Daniel Liberzon. Stability of Hybrid Automata with Average Dwell Time: An Invariant Approach. Submitted to *IEEE Conference on Decision and Control*, 2004.

[93] Athicha Muthitacharoen, Seth Gilbert, and Robert Morris. Atomic Mutable Data in a Distributed Hash Table with Etna. Submitted for publication.

In Progress

[94] Paul C. Attie and Nancy A. Lynch. A compositional automaton-based model for dynamic systems. In progress.

[95] Indraneel Chakraborty, Rui Fan, Nancy Lynch, and Boaz Patt-Shamir. Clock synchronization for ad-hoc wireless networks. In progress.

[96] M. Demirbas, A. Arora, T. Nolte, and N. Lynch. Self-stabilizing hierarchical tracking service for sensor networks. In progress.

[97] Stephen Garland and Nancy Lynch, Joshua Tauber, and Mandana Vaziri. IOA user guide and reference manual. In progress.

[98] Joshua Tauber. Verifiable Code Generation from Abstract I/O Automata Models for Distributed Computing. Manuscript, March 2001. (Thesis Proposal)

[99] Joshua A. Tauber and Stephen J. Garland. Definition and Expansion of Composite Automata in IOA. Manuscript, In progress.

[100] Joshua Tauber and Stephen Garland. Expansion of composite automata into primitive automata. In preparation.

Other References

[1] Y. Amir, D. Dolev, P. Melliar-Smith, and L. Moser. Robust and efficient replication using group communication. Technical Report CS94-20, Institute of Computer Science, Hebrew University, Jerusalem, Israel, 1994.

[2] Myla Archer and Constance Heitmeyer. Verifying hybrid systems modeled as timed automata: A case study. In Oded Maler, editor, *Hybrid and Real-Time Systems* (International Workshop, HART'97, Grenoble, France, March 1997), volume 1201 of *Lecture Notes in Computer Science*, pages 171–185, Berlin Heidelberg, 1997. Springer-Verlag.

[3] Hagit Attiya, Amotz Bar-Noy, and Danny Dolev. Sharing memory robustly in message-passing systems. *Journal of the ACM*, 42(1):124–142, January 1995.

[4] Ö. Babaoğlu, R. Davoli, L. Giachini, and M. Baker. Relacs: A communication infrastructure for constructing reliable applications in large-scale distributed systems. TR UBLCS94-15, Laboratory of Computer Science, University of Bologna, 1994.

[5] K. P. Birman and R. van Renesse. *Reliable Distributed Computing with the Isis Toolkit.* IEEE Computer Society Press, Los Alamitos, CA, 1994.

[6] Anna Chefter A Simulator for the IOA Language. Master of Engineering and Bachelor of Science in Computer Science and Engineering Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, May, 1998.

[7] Gregory V. Chockler. An adaptive totally ordered multicast protocol that tolerates partitions. Master's thesis, Institute of Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel, August 1997.

[8] Gregory Chockler, Idit Keidar, and Roman Vitenberg. Group communication specifications: A comprehensive study. *ACM Computing Surveys*, 33(4):1–43, December 2001. Previous version: MIT Technical Report MIT-LCS-TR-790, September 1999.

[9] Roberto DePrisco. *On Building Blocks for Distributed Systems.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, December 1999. Also, Technical Report MIT/LCS/TR-803, MIT Laboratory for Computer Science, Cambridge, MA.

[10] R. De Prisco, A. Fekete, N. Lynch, and A.A. Shvartsman. A dynamic primary configuration group communication service. In Prasad Jayanti, editor, *Distributed Computing* Proceedings of DISC'99 - 13th International Symposium on Distributed Computing, Bratislava, Slovak Republic, September 1999, volume 1693 of *Lecture Notes in Computer Science*, pages 64–78, Bratislava, Slovak Republic, 1999. Springer-Verlag-Heidelberg.

[11] Shlomi Dolev, Roberto Segala, and Alex Shvartsman. Dynamic load balancing with group communication. In *6th International Colloquium on Structural Information and Communication Complexity (SIROCCO'99)*, Bordeaux, France, 1999.

[12] D. Dolev and D. Malki. The Transis approach to high availability cluster communications. *Communications of the ACM*, 39(4):64–70, 1996.

[13] Michael D. Ernst, Jake Cockrell, William G. Griswold, and David Notkin. Dynamically discovering likely program invariants to support program evolution. *IEEE Transactions on Software Engineering*, 27(2):1–25, February 2001. safety/Ernst-invariants-tse.pdf.

[14] P. D. Ezhilchelvan, A. Macedo, and S. K. Shrivastava. Newtop: a fault tolerant group communication protocol. In *15th International Conference on Distributed Computing Systems (ICDCS)*, June 1995.

[15] Alan Fekete, Nancy Lynch, and Alex Shvartsman. Specifying and Using a Partitionable Group Communication Service. *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 53-62, Santa Barbara, CA, August 1997. Expanded version in Technical Memo.

[16] Alan Fekete, Nancy Lynch, and Alex Shvartsman. Specifying and Using a Partitionable Group Communication Service. Technical Memo MIT-LCS-TM-570b, MIT Laboratory for Computer Science, Cambridge, MA, 1999. Later version in [5].

[17] Stephen Garland and John Guttag, A guide to LP, the Larch Prover. Technical report, DEC Systems Reserach Center, 1991. Updated version avaliable at URL `http://nms.lcs.mit.edu/Larch/L`

[18] C. Georgiou and A. Shvartsman. Cooperative computing with fragmentable and mergeable groups. In *Proc. of 7th International Colloquium on Structure of Information and Communication Complexity SIROCCO'00*, page 15pp, 2000.

[19] Mark G. Hayden. *The Ensemble System.* PhD thesis, Department of Computer Science, Cornell University, January 1997.

[20] Jason Hickey, Nancy Lynch, and Robbert van Renesse. Specifications and proofs for Ensemble layers. In Rance Cleaveland, editor, *Tools and Algorithms for the Construction and Analysis of Systems, Fifth International Conference, (TACAS'99), Amsterdam, the Netherlands, March 1999*, volume 1579 of *Lecture Notes in Computer Science*, pages 119–133. Springer-Verlag, 1999.

[21] I. Keidar and D. Dolev. Efficient message ordering in dynamic networks. In *15th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 68–76, May 1996.

[22] Roger Khazan, Alan Fekete, and Nancy Lynch. Multicast group communication as a base for a load-balancing replicated data service. In *12th International Symposium on Distributed Computing*, pages 258–272, Andros, Greece, September 1998.

[23] L. Lamport. The part-time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998. Earlier version in Research Report 49, Digital Equipment Corporation Systems Research Center, Palo Alto, CA, September 1989.

[24] N. Lesley and A. Fekete. Providing view synchrony for group communication services. In *Proceedings of the Australian Computer Science Conference*, pages 457–468, Auckland, New Zealand, January 1999.

[25] L. Li, J. Halpern, V. Bahl, Y.M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *ACM Symposium on Principle of Distributed Computing (PODC)*, August 2001.

[26] E. Yeger Lotem, I. Keidar, and D. Dolev. Dynamic voting for consistent primary components. In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 63–71, Santa Barbara, CA, August 1997.

[27] Nancy Lynch. Algorithms for Data Sharing, Coordination, and Communication in Dynamic Network Settings. AFOSR Contract, 2004.

[28] Nancy Lynch and Alex Shvartsman. Robust emulation of shared memory using dynamic quorum-acknowledged broadcasts. *Twenty-Seventh Annual International Symposium on Fault-Tolerant Computing (FTCS'97)*, pages 272-281, Seattle, Washington, June 1997.

[29] Robin Milner. *Communicating and mobile systems: the Pi-Calculus.* Part I. Cambridge University Press, United Kingdom, 1999.

[30] Robin Milner. *Communicating and mobile systems: the Pi-Calculus.* Part II. Cambridge University Press, United Kingdom, 1999.

[31] L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, R. K. Budhia, and C. A. Lingley-Papadopoulos. Totem: A fault-tolerant multicast group communication system. *Communications of the ACM*, 39(4), April 1996.

[32] Paul Pettersson and Kim G. Larsen. Uppaal2k. *Bulletin of the European Association for Theoretical Computer Science*, 70:40–44, 2000.

[33] Proceedings of the First international Workshop on Peer-to-Peer Systems (IPTPS02), March 2002.

[34] Radio Technical Commission for Aeronautics. Minimum operational performance standards for TCAS airborn equipment. Technical Report RTCA/DO-185, RTCA, September 1990. Consolidated Edition.

[35] Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1995. Also, MIT/LCS/TR-676.

[36] Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, August 1995. Special issue from [37].

[37] Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. In Bengt Jonsson and Joachim Parrow, editors, *CONCUR'94: Concurrency Theory* (5th International Conference, Uppsala, Sweden, August 1994), volume 836 of *Lecture Notes in Computer Science*, pages 481–496. Springer-Verlag, 1994. A revised version appears in [36].

[38] Marïelle Stoelinga. *Alea jacta est: Verification of Probabilistic, Real-time and Parametric Systems.* PhD thesis, University of Nijmegen, the Netherlands, April 2002.

[39] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM 2001*, pages 149–160, San Deigo, CA, August 2001.

[40] Qixiang Sun. Reliable multicast for publish/subscribe systems. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, May 2000.

[41] J. Sussman, I. Keidar, and K. Marzullo. Optimistic virtual synchrony. Technical Report MIT-LCS-TR-792, MIT Lab for Computer Science, November 1999. Also Technical Report CS1999-634 University of California, San Diego, Department of Computer Science and Engineering.

[42] R. van Renesse, K. P. Birman, and S. Maffeis. Horus: A flexible group communication system. *Communications of the ACM*, 39(4):76–83, 1996.

[43] Robbert van Renesse, Ken Birman, Mark Hayden, Alexey Vaysburd, and David Karr. Building adaptive systems using Ensemble. *Software-Practice and Experience*, 29(9):963–979, July 1998.

7 Interactions/Transitions

7.1 Participation/presentations at meetings, conferences, seminars, etc.

Paul Attie Dynamic Input/Output Automata: A Formal Model for Dynamic Systems. CONCUR'01: 12th International Conference on Concurrency Theory, Aalborg, Denmark, August 2001.

Omar Bakr Evaluating the Running Time of a Communication Round over the Internet. PODC'02, Monterey, CA, July, 2002

Ziv Bar-Joseph: Early-Delivery Dynamic Atomic Broadcast. 16th International Symposium on DIStributed Computing (DISC), Toulouse, France, October 2002.

Rui Fan: Efficient Replication of Large Data Objects. Twenty-Second Annual ACM Symposium on Principles of Distributed Computing (PODC), page 335, Boston, MA, July, 2003.

Seth Gilbert: GeoQuorums: Implementing Atomic Memory in Ad Hoc Mobile Networks. DISC 2003, Sorrento, Italy, October 2003.

GeoQuorums: Implementing Atomic Memory in Ad Hoc Mobile Networks. Oxygen Student Workship, Sept. 2003,

Rambo II: Rapidly Reconfigurable Atomic Memory for Dynamic Networks. International Conference on Dependable Systems and Networks (DSN), San Francisco, CA, June 2003.

Dilsun Kirli Kaynar Simulating Nondeterministic Systems With Multiple Levels of Abstraction. Tools Day, a satellite event of CONCUR 2002 in Brno, Czech Republic, August, 2002

Idit Keidar On the Cost of Fault-Tolerant Consensus When There Are No Faults - A Tutorial. PODC'02, Monterey, CA, July, 2002.

Challenges in Evaluating Distributed Algorithms. International Workshop on Future Directions in Distributed Computing (FuDiCo), Bertinoro, Italy, June, 2002.

On the Performance of Consensus Algorithms: Theory and Practice. EPFL, Switzerland, Communication Systems Department Seminar Series and the Technion CLUBNET Networking Seminar Series, March, 2002.

Availability Study of Dynamic Voting Algorithms. 21st International Conference on Distributed Computing Systems (ICDCS), Phoenix, Arizona, April 2001.

A Framework for Highly Available Services Based on Group Communication. IEEE International Workshop on Applied Reliable Group Communication (WARGC), held in conjunction with ICDCS 2001,Phoenix, Arizona, April 2001.

Roger Khazan An Algorithm for an Intermittently Atomic Data Service Based on Group Communication. International Workshop on Large-Scale Group Communication, Florence, Italy, October 2003.

Incremental Modeling and Verification; and Group Communication. Northeastern U. March, 2003.

Incremental Modeling and Verification; and Group Communication. Lincoln Labs, February, 2002.

Incremental Modeling and Verification; and Group Communication. UConn, December 2001.

Roger Khazan was invited by Draper Laboratories to give an introductory 2-hour lecture about Distributed Algorithms and our group's work on July 19, 2001.

A Client-Server Approach to Virtually Synchronous Group Multicast: Specifications and Algorithms. IEEE 20th International Conference on Distributed Computing Systems (ICDCS), Taipei, Taiwan, April 2000.

An Inheritance-Based Technique for Building Simulation Proofs Incrementally. ICSE, Limerick, Ireland, June 2000.

Carolos Livadas: Modeling, Analyzing, and Improving SRM; A Formal Venture into the Realm of Reliable Multicast. Cambridge Research Laboratory, HP Laboratories, June 16, 2003.

Modeling, Analyzing, and Improving SRM; A Formal Venture into the Realm of Reliable Multicast. BBN Technologies, Internetwork Research and Mobile Systems Departments, May 7, 2003.

Modeling, Analyzing, and Improving SRM; A Formal Venture into the Realm of Reliable Multicast. Lincoln Laboratories, April 11, 2003.

A Formal Venture into Reliable Multicast Territory. Formal Techniques for Networked and Distributed Systems - FORTE 2002, Houston, Texas, November 2002.

Designing a Caching-Based Reliable Multicast Protocol. DSN'01 in Sweden.

Nancy Lynch: Reconfigurable Atomic Memory for Dynamic Networks. Ecole Polytechnique, Lausanne, France, June 2003.

RAMBO: A Reconfigurable Atomic Memory Service for Dynamic Networks. Boston University, April 2003.

Hybrid Input/Output Automata: Theory and Applications. 19th Conference on the Mathematical Foundations of Programming Semantics, March 2003.

RAMBO: A Reconfigurable Atomic Memory Service for Dynamic Networks. Grace Hopper Distinguished Lecturer, University of Pennsylvania, February 2003.

RAMBO: A Reconfigurable Atomic Memory Service for Dynamic Networks. University of California, San Diego, January 2003.

RAMBO: A Reconfigurable Atomic Memory Service for Dynamic Networks. 16th International Symposium on DIStributed Computing (DISC), Toulouse, France, October 2002.

RAMBO: A Reconfigurable Atomic Memory Service for Dynamic Networks. Distinguished Lecturer, Johns Hopkins University, September 2002.

Modeling Distributed Systems Using I/O Automata. Draper Labs, August 30, 2002.

New Directions for NEST Research. Building Block for High-Performance, Fault-Tolerant Distributed Systems, NEST Darpa PI meeting, July 12, 2002.

Impossibility of Consensus With One Faulty Process. High school women's summer program, MIT, July 1, 2002.

Early-Delivery Dynamic Atomic Broadcast. U. Salerno, Salerno, Italy, June 10, 2002.

Building Blocks for High Performance, Fault-Tolerant Distributed Systems. AFOSR PI meeting, June 3, 2002.

Hybrid I/O Automata, MIT SEC meeting, May 17, 2002.

Early-Delivery Dynamic Atomic Broadcast, NYU, May 3, 2002.

Early-Delivery Dynamic Atomic Broadcast. U. Nijmegen, April 22, 2002.

Research Directions in Computer System Security, NSF Workshop, February 14, 2002.

Early-Delivery Dynamic Atomic Broadcast. UC Berkeley, February 12, 2002.

Early-Delivery Dynamic Atomic Broadcast. Tufts University, February 6, 2002.

Implementing Atomic Objects in a Dynamic Environment. Leslie Lamport's 60th Birthday Celebration. PODC'01, Newport, R.I. August, 2001.

Dynamic Input/Output Automata: a Formal Model for Dynamic Systems. PODC'01, Newport, R.I. August, 2001.

Hybrid I/O Automata, a Mathematical Model for Hybrid Systems. Albert Meyer's 60th Birthday Celebration. Boston, MA. June, 2001.

Hybrid I/O Automata, a Mathematical Model for Hybrid Systems. Laboratory for Information and Decision Systems, MIT. May, 2001.

Totally Ordered Multicast with QoS. Workshop on Perspectives on Algorithms and Distributed Algorithms. Luminy, France. May, 2001.

Hybrid I/O Automata, a Mathematical Model for Hybrid Systems. Workshop on Perspectives on Algorithms and Distributed Algorithms. Luminy, France. May, 2001.

Hybrid I/O Automata, a Mathematical Model for Hybrid Systems. University of Pennsylvania, Phila., PA. April, 2001.

Hybrid Input/Output Automata, Revisited, Hybrid Systems: Computation and Control. Rome, Italy. March, 2001.

Defining the Oxygen Software Architecture. Oxygen brainstorm meeting, MIT. February, 2001.

Hybrid Input/Output Automata, Revisited, Hybrid Systems: Computation and Control. Rome, Italy. March, 2001.

Reliable Group Communication: A Mathematical Approach. Distinguished Lecture. Purdue University. November, 2000.

Nancy Lynch and Stephen Garland: Modeling and Analyzing Distributed Systems using I/O Automata. Draper R&D Kickoff, August 2002.

Sayan Mitra: Safety Verification of Model Helicopter Controller using Hybrid Input/Output Automata. 6th International Workshop, HSCC'03, Prague, the Czech Republic April 3-5, 2003.

David Ratajczak Atomic Data Access in Content-Addressable Networks. MIT Workshop on Peer-to-Peer Computing, March, 2002.

Alex Shvartsman Distributed Cooperation in the Presence of Failures and Delays. Computer Science Colloquium, Yale University, 2000.

Distributed Cooperation in the Presence of Failures and Delays, Computer Science Seminar, Ecole Polytechnique, France, 2000.

7.2 Consultative and advisory functions to other laboratories and agencies, especially Air Force and other DoD laboratories

See "Transitions", below, for further details about these projects.

Nancy Lynch and Stephen Garland: Assisted developers at Draper Laboratories in a project to model and analyze a critical military system. Fall, 2001-Spring, 2003. Contact: Joe Kochocki.

Nancy Lynch: Occasional contact with Lincoln Labs, through AFOSR-funded ex-PhD student Roger Khazan. Working on design of a chat-like communications application for air force missions. In progress.

Nancy Lynch: Consultant to director of the Division of Computer and Information Sciences and Engineering, NSF. September 2000-June 2002 Contact: Dr. Ruzena Bajcsy, NSF.

Nancy Lynch: Technical advisor, Centrata corporation.

Sayan Mitra: Working at Naval Research Laboratory summer 2003, with Dr. Myla Archer. Developing tools for modeling and analyzing timing-based and hybrid systems of use in the Navy.

7.3 Transitions

Transition 1:

(a) Customer:

Center for High Assurance Computer Systems Naval Research Laboratory Code 5546 4555 Overlook Avenue, S.W. Washington, DC 20375-5320

Dr. Myla Archer. archer@itd.nrl.navy.mil, 202-767-2389

Dr. Connie Heitmeyer. Head, Software Engineering. heitmeyer@itd.nrl.navy.mil, 202-767-3596

(b) Research result:

The definition of our timed I/O automaton mathematical model. Methods of modeling timing constraints. Inductive proof methods for proving invariant assertions, simulation relationships, and timing properties. The new definitions of hybrid I/O automata.

(c) Application:

Developers at the NRL have used our timed I/O automaton model and its proof methods as the basis of the TAME tool for modeling and verifying high assurance systems of interest to the Navy.

One of my PhD students, Sayan Mitra, worked at NRL last summer, developing the tools further, in particular, adding support for hybrid systems.

(d) What was accomplished:

This has led to usable modeling and proof tools. It has also led to improvements in the PVS theorem-prover, which is used by the TAME system for carrying out formal proofs. Various application case studies have been carried out.

(e) Why it is important:

Timed and hybrid I/O automata form a sound mathematical foundation for distributed systems with timing constraints. Sound methods for describing and analyzing the design of such systems require such a foundation. Such methods can be used to make the process of developing distributed systems more efficient and reliable.

Transition 2:

(a) Customer:

Nippon Telephone and Telegraph Communication Science Labs 2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan, 619-0237

Ken Mano, mano@cslab.kecl.ntt.co.jp

Yoshifumi Manabe, manabe@cslab.kecl.ntt.co.jp

(b) Research result:

Our modeling and verification techniques for distributed systems, based on (untimed) I/O automata. Our formal IOA language and its proof tools.

(c) Application:

Researchers at NTT have used our model, techniques, language, and tools to describe the agent programming systems that they are building. In particular, they are applying our results to the design and implementation of their general NePi2 agent programming system.

(d) What was accomplished:

One NTT employee, Yoshinobu Kawabe, produced a complete model of the NePi2 system implementation, in several levels of abstraction, using IOA. Furthermore, he carried out a complete proof of correctness, using invariants and simulation relations, using our theorem-proving tools.

(e) Why it is important:

Besides what was accomplished for the specific NePi2 system, this work demonstrates the feasibility of validating complete, significant-sized system designs using our methods.

Transition 3:

(a) Customer: Draper Laboratories 555 Technology Square Cambridge, MA 02139

Joseph Kochocki, jkochocki@draper.com, 617-258-1285

(b) Research result:

Our modeling and verification techniques for distributed systems with timing constraints, based on timed I/O automata. In particular, our methods for decomposing systems into interacting components and into levels of abstraction, and our methods of modeling time deadlines for events.

(c) Application:

We worked directly with Draper Labs software developers and management, performing a thorough analysis of a simulated version of a major system that Draper is developing. MEng student Vida Ha, working as a Draper Fellow, carried out the bulk of the work.

(d) What was accomplished:

A complete model for the system, at multiple levels of abstraction, was developed. Statements of key properties satisfied at the various levels, were produced. These properties include critical system reliability and timing properties. Formal proof sketches of some of the statements were carried out.

(e) Why it is important:

Systems of this kind are critically important to the national interest. Yet the design and implementation techniques used currently are unwieldy and unreliable. The key to making the development process more tractable and more reliable is to raise the level of abstraction at which the systems are described and analyzed.

Transition 4:

(a) Customer: MIT Department of Aeronautics and Astronautics

Prof. Eric Feron, feron@mit.edu, 617-253-1991

(b) Research result: Our mathematical model for hybrid continuous/discrete systems, which we call Hybrid I/O Automata. Our inductive methods for proving properties of hybrid systems. Our proposed formal language for describing hybrid I/O automata.

(c) Application: The Aero/Astro department uses a Quanser model helicopter in teaching students how to design helicopter controllers. In order to protect the model from badly-designed controllers, developers in Aero/Astro needed to add a "supervisory controller" module to their system. We worked with Aero/Astro developers and researchers to develop a design for such a controller. We helped to document the design in terms of the HIOA model, and helped to prove it safe using our proof methods.

(d) What was accomplished: The design was completely developed and validated. Implementation of the actual controller for the physical helicopter system is nearly completed.

(e) Why it is important: This demonstrates the feasibility of complete modeling and analyses of such systems, at high levels of abstraction. Such models and proofs provide high assurance, ahead of time, that the systems will work as intended. They also make explicit the precise reasons why the system behaves correctly. Such models can be reused to help in development of similar systems.

Transition 5:

(a) Lincoln Laboratories

Dr. Roger Khazan, roger@lcs.mit.edu, 781-981-5976

Dr. Cliff Weinstein, cjw@ll.mit.edu 781-981-7621

(b) Research result: Algorithms and service definitions for group communication services. Techniques for designing and analyzing such algorithms and services.

(c) Application: Roger Khazan, who worked on our AFOSR project at MIT, joined Lincoln Labs in 2002 as a research staff member. His PhD thesis was on group communication algorithms and services. He has started a new effort at Lincoln Labs to develop group communication services for military use, specifically, for "chat-style" services to aid in communication during missions. Current communication services that the new ones are designed to replace have problems with reliability and efficiency; it is hoped that the application of group communication technologies will result in more robust, more efficient communication systems.

(d) What was accomplished: Preliminary design discussions are under way. They hope to develop a credible design of use to the military.

(e) Why it is important: If the new project is successful, it will result in more robust, more efficient communication systems for military missions.

8 New discoveries, inventions, patent disclosures

One patent was issued: "Model-Based Software Design and Validation" by Stephen Garland and Nancy Lynch, Sept. 11, 2001. This is for our work on the IOA language and tools, specifically for a design methodology combining theorem-proving and code generation, for distributed systems.

9 Honors and awards

Toh Ne Win: Received an award from the MIT EECS department last spring for the best MEng thesis in computer science, 2003.

Seth Gilbert: His paper with Nancy Lynch, Alex Shvartsman, and Jennifer Welch, "GeoQuorums: Implementing atomic memory in ad hoc networks," was selected for special edition of *Distributed Computing*, 2003.

Stephen Garland and Nancy Lynch: IOA work was featured as one of Technology Review's "10 Emerging Technologies That Will Change the World, " February, 2003.

Carl Livadas: Chosen as a Barger Fellow at BBN (named after Dr. James Barger who is a distinguished research scientist at BBN), 2003. This is a new Fellowship program at BBN that is used to attract distinguished new PhD graduates. Carl was chosen as the first such fellow based on the work he did in the TDS group.

Nancy Lynch: Chosen as Grace Hopper lecturer, U. Pennsylvania, 2003.

Gregory Chockler: His paper with Dahlia Malkhi "Active Disk Paxos with Infinitely Many Processes," was selected to appear in the PODC 2002 issue of the Distributed Computing journal.

Roger Khazan: His paper with Idit Keidar, Roger Khazan, Nancy Lynch and Alex Shvartsman, "An Inheritance-Based Technique for Building Simulation Proofs Incrementally," was invited for submission to *TOSEM*, 2002.

Nancy Lynch: Distinguished Lecturer, Johns Hopkins University, 2002.

Chris Luhrs: Winner of the Seventh Annual Anna Pogosyants UROP prize for his work with Stephen Garland and Nancy Lynch, 2002.

Andrej Bogdanov: Honorable mention for MEng thesis, 2001.

Idit Keidar: won the Alon Fellowship for Junior Faculty, 2001.

Idit Keidar: awarded a Technion Management Career Development Chair, 2001.

Nancy Lynch: Elected Member of National Academy of Engineering, 2001; ACM Fellow.

Nancy Lynch, together with Michael Fischer and Michael Paterson: won the second annual Principles of Distributed Computing (PODC) conference award for "most influential paper in the field", 2001. They won this award for their 1985 paper "Impossibility of Distributed Consensus with one Nonfaulty Process".

Alex Shvartsman: won the Outstanding Research Award for Junior Faculty at the University of Connecticut, 2001.

Nancy Lynch: Distinguished Lecturer, Purdue, University, 2000.

Michael Tsai: Winner of the Fifth Annual Anna Pogosyants UROP Award for his work on the IOA Toolset with Joshua Tauber and Nancy Lynch, 2000.

Gregory Chockler: His paper with Danny Dolev, Roy Friedman and Roman Vitenberg, "Implementing Caching Service for Distributed CORBA Objects," won two best paper awards in the Middleware 2000 Conference.